**AVACADO DATASET**

I was given a project to design prediction models of Avacado Dataset using Machine Learning. The prediction of Avacado Price is the main intent to design this model using other features of the dataset.

|  |  |
| --- | --- |
| Avacado Dataset consists of the prices of avacados in different areas at different times in USA. The Product Look Up(PLU) Codes are only for Hass avacados. 4046,4225,4770 are the PLUs showing the sales volume of avacados.  There are several features in the dataset. |  |

**Description:**

* **Unnamed** – It’s like indexing
* **Date** - The date of the observation
* **AveragePrice** - the average price of a single avocado
* **type** - conventional or organic
* **year** - the year
* **region** - the city or region of the observation
* **Total Volume** - Total number of avocados sold
* **4046** - Total number of avocados with PLU 4046 sold
* **4225** - Total number of avocados with PLU 4225 sold
* **4770** - Total number of avocados with PLU 4770 sold

**Problem Definition:**

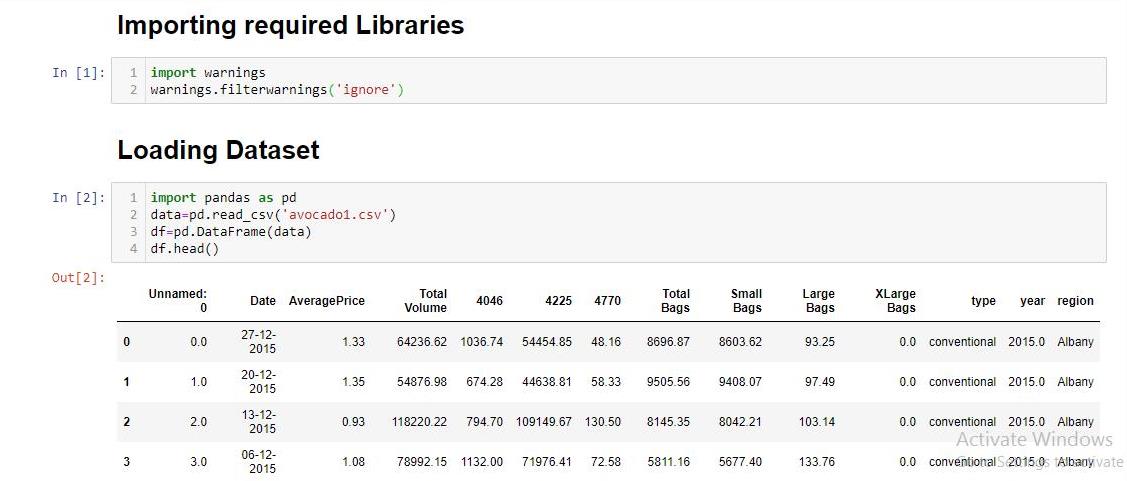
In the Dataset, there are two such features which can be taken as target variable that can be predicted by modeling the dataset. Those are **AveragePrice** and **region**.

**AveragePrice** is a numerical and continuous data feature which indicate average price of avacados in the dataset. So It can be predicted by a Regression Model .

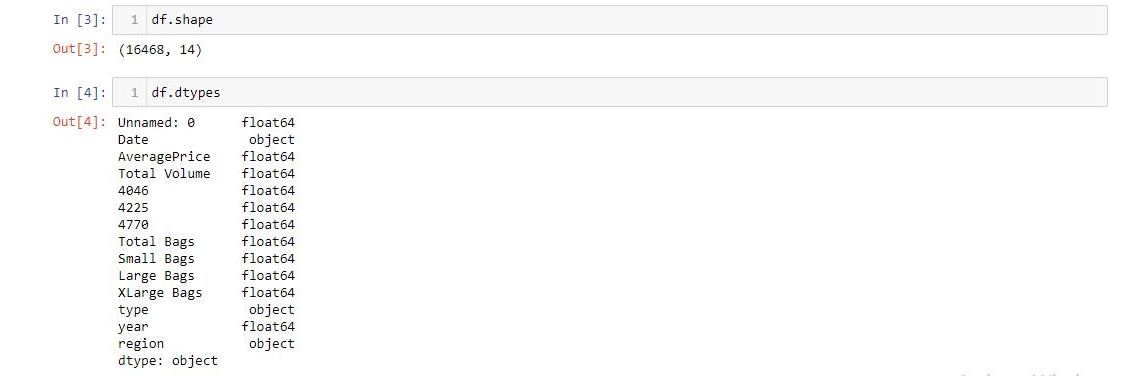
**region** is a categorical data feature which describes the areas in the dataset. It can be predicted by a Classification Model.

**Data Analysis:**

In the Jupyter Notebook, First of all Warnings library is used to ignore the warnings so that the programmers or viewers won’t be confused. Then the avocado dataset is uploaded by using pandas library code pd.read\_csv as the dataset is a csv file.

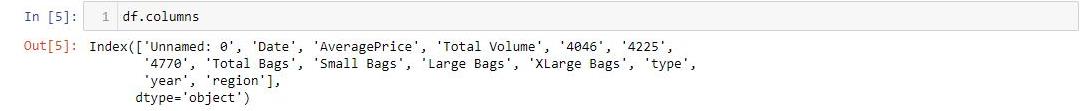


The dataset has 14 features. The shape and type of the dataset is determined using df.shape and df.types respectively where df is the DataFrame of the avocado dataset.

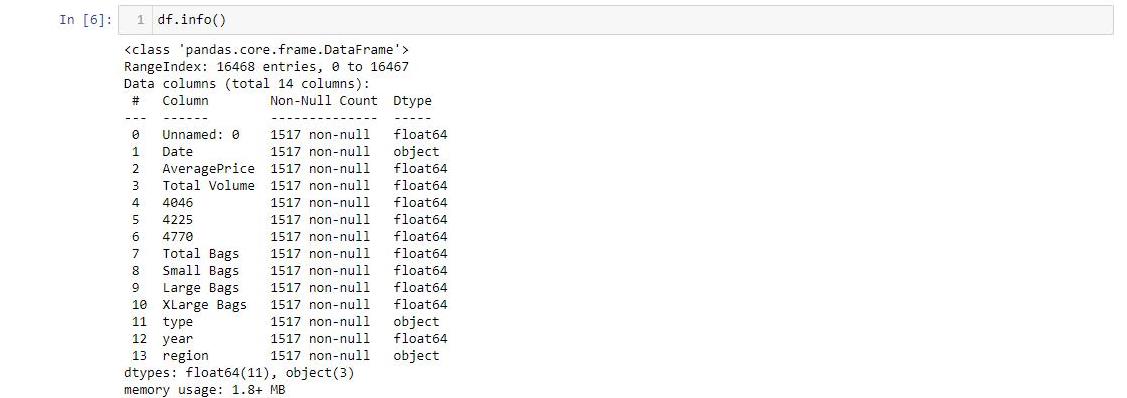


Date, type and region are object type and rest all are in float type.

df.columns is used to list the features/columns of the dataset.

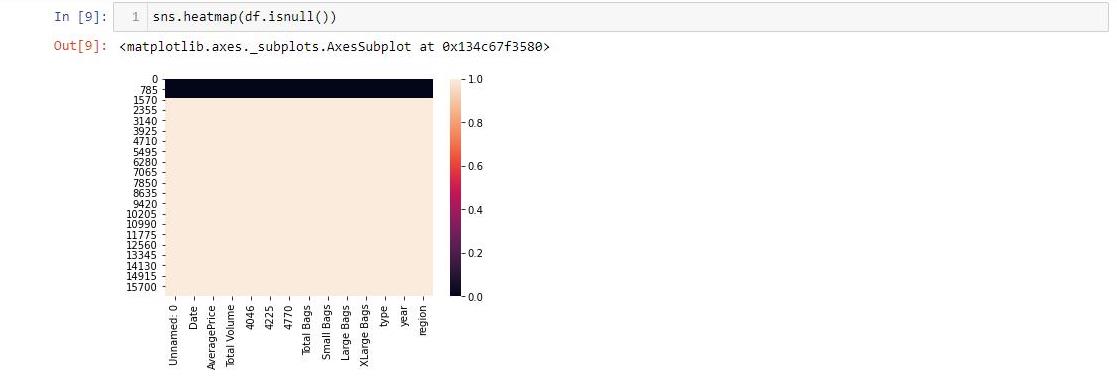


df.info shows 1517 rows of the dataset is not null.



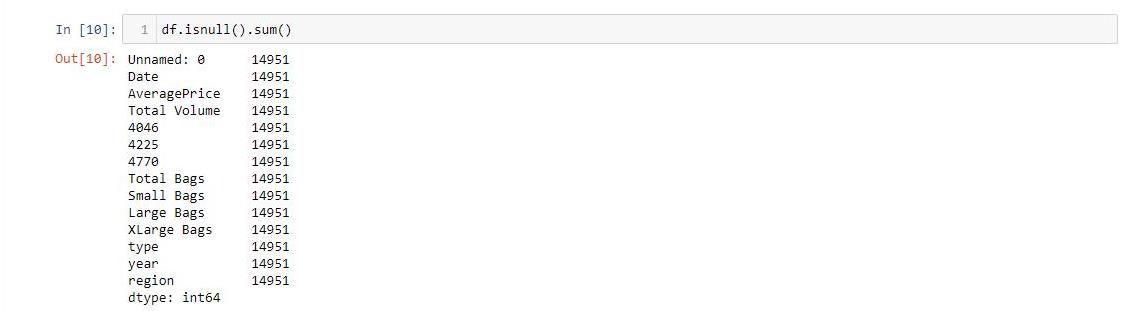
Importing Numpy, Seaborn and Matplotlib for **Exploratory Data Analysis(EDA)**. First command is used to check null values in the dataset.





There are so much null values in the dataset.

14951 rows are null means blank in the dataset.



The number of non null values is 1517. The df will be again updated with those 1517 rows.

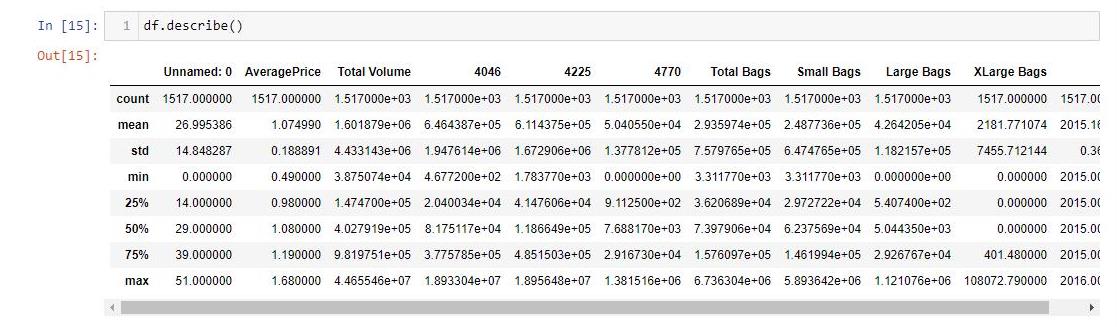


Now, the shape of the dataframe became 1517 rows and 14 columns.



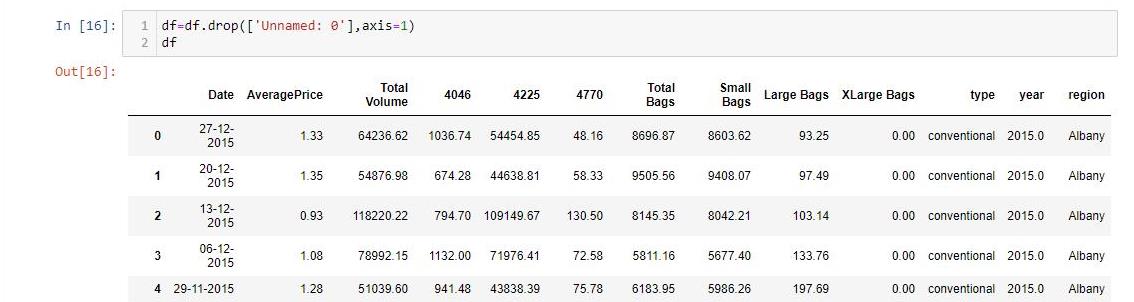
To describe the dataframe, df.describe is used to show the minimum, maximum values of the dataframe for each column data.

It also gives the 25th,50th,75th percentiles and mean & standard deviation of the dataframe. 50th percentile is known as median.



In Volumes and Bags columns the standard deviation is higher.

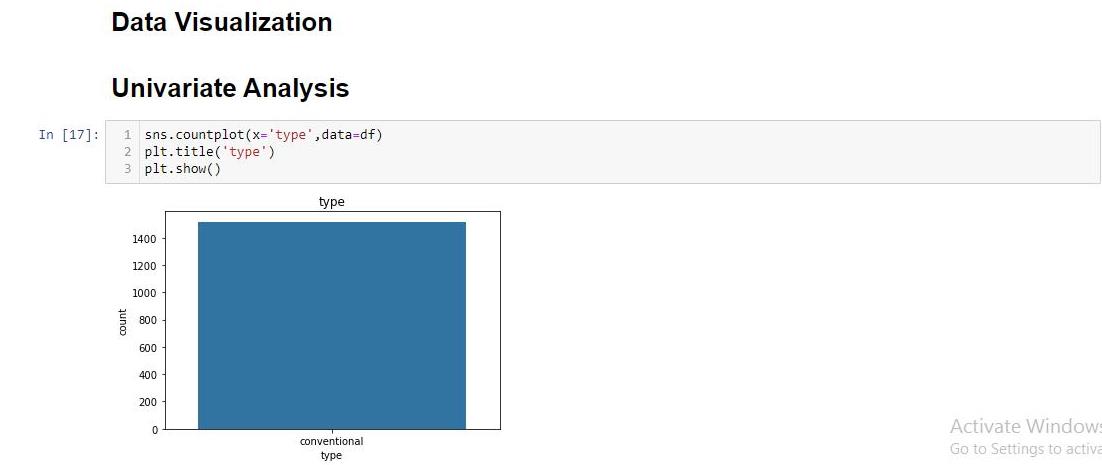
Unnamed column is like a index column. It will be dropped.



Now, the shape of the dataframe became 1517 rows and 13 columns.

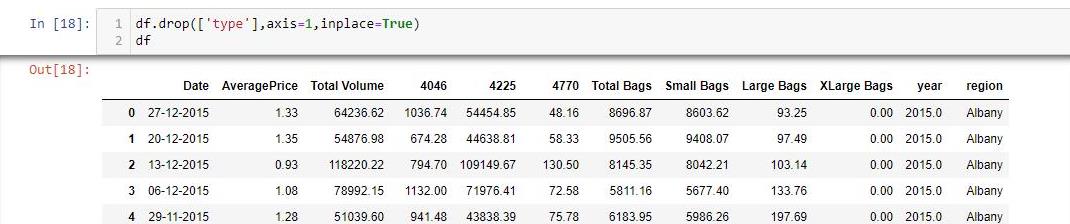
**Data Visualization** is the major part of EDA process. It shows the graphical representation of the features and relationship between them.

***Univariate Analysis*** is used to represent one feature’s distribution in the dataset and the counts for a categorical feature.



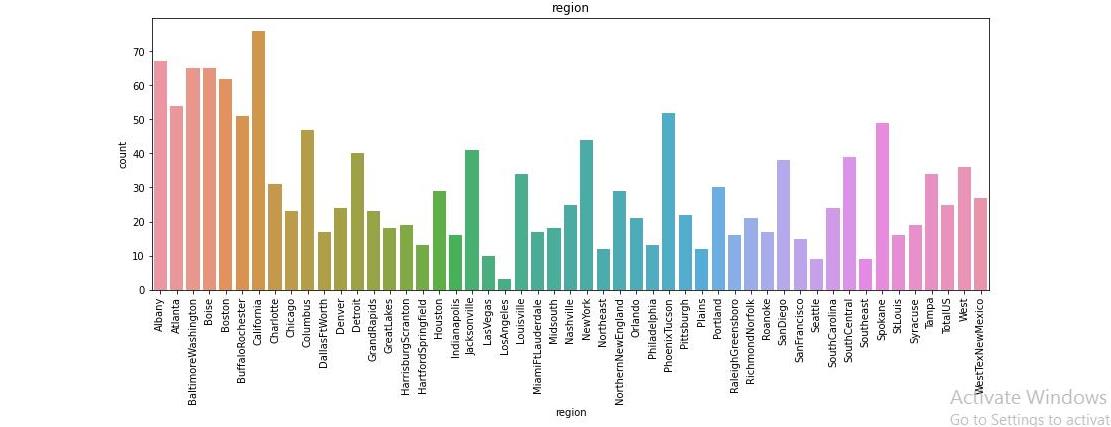
The univariate analysis of ‘type’ feature shows that in the whole dataframe there is only one type ie conventional. So the value of this feature is one or unique for the whole dataframe(dataset). It is now for no use for building prediction model.

That’s why ‘type’ feature will be dropped.



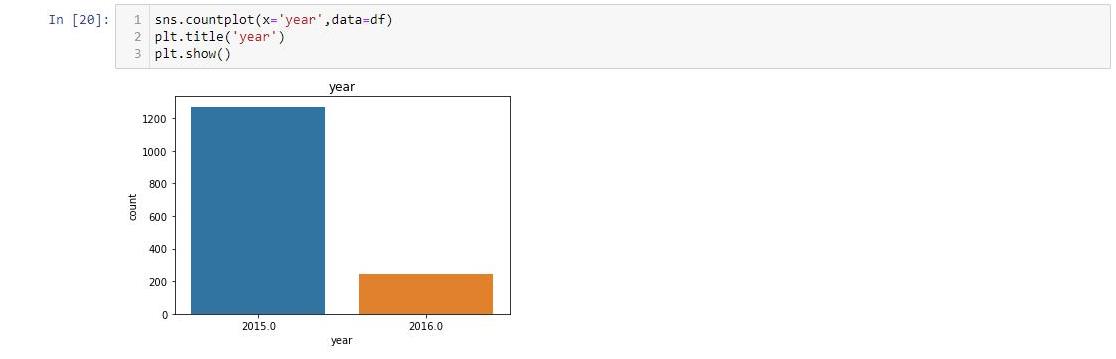
Now again the shape is reduced and the dataframe shape became 1517 rows and 12 columns.

The univariate analysis of region shows different types of areas in which the avocado are sold. The bars of this graph shows the number of sales in a particular region.



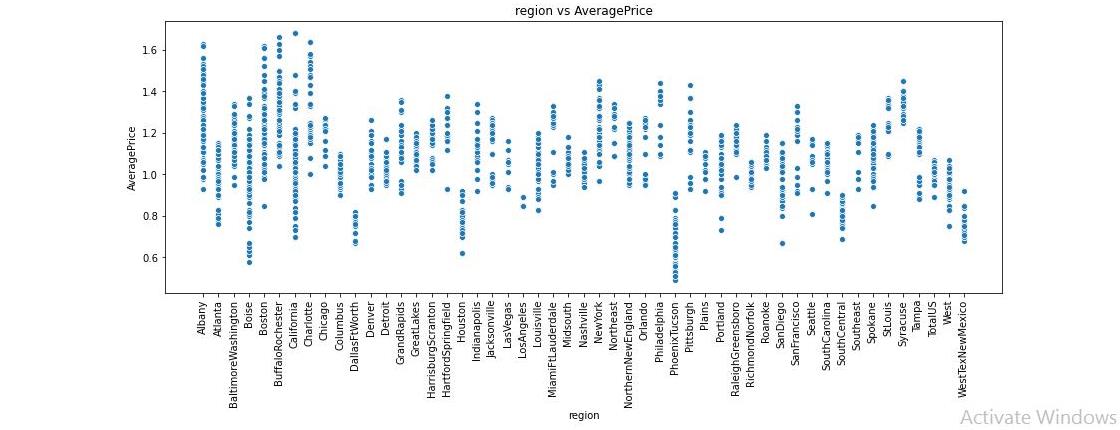
* California had highest sales of avocado.

The univariate analysis of year shows most datas in the dataset is taken in 2015.



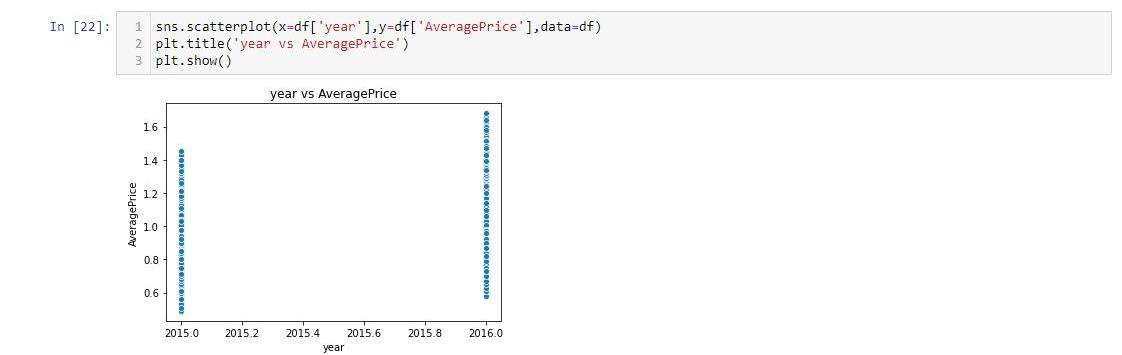
***Bivariate Analysis*** shows the graphical representation of the relationship between two features in the dataset.

Let’s check the relationship between region and AveragePrice.



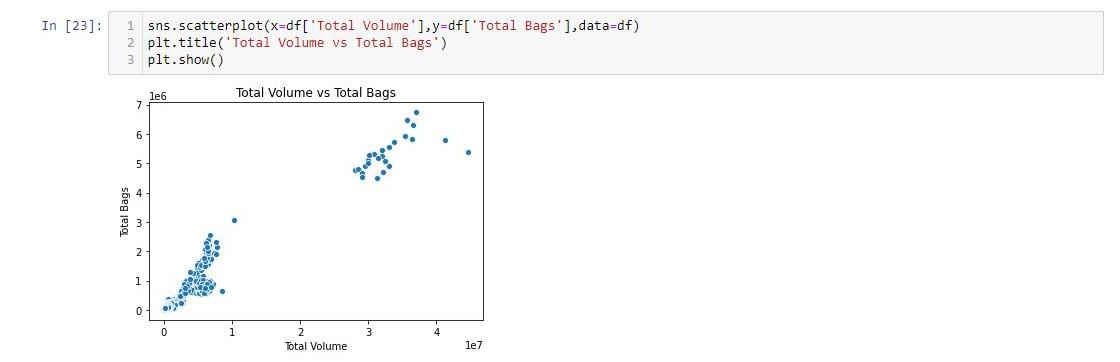
* Region and Average price are having some relationship that varies.
* PhoenixTuscon is having less Average Price whereas California is having highest Average Price.

The Bivariate Analysis between Year and AveragePrice.

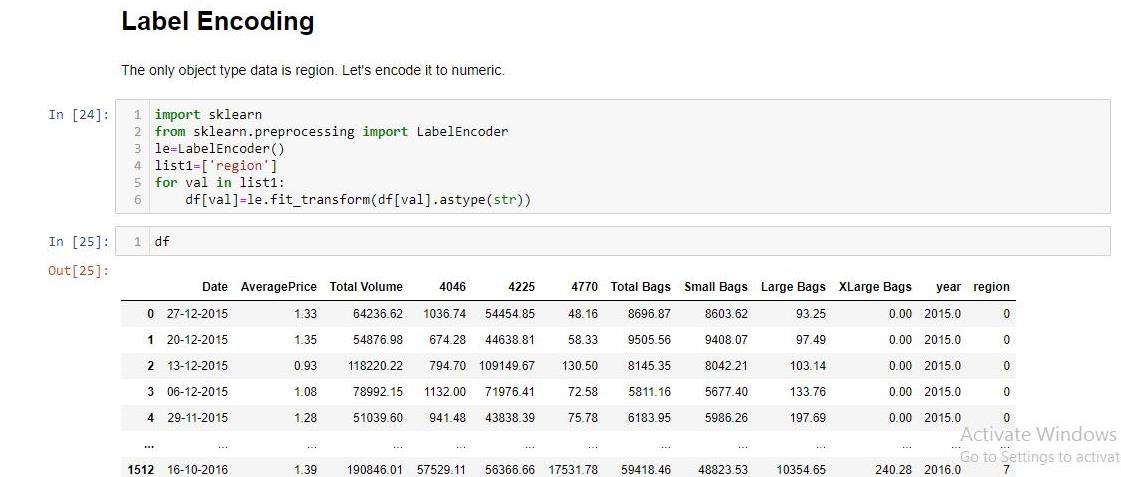


* In the year 2016, The Avacado price is increased.

The bivariate analysis between Total Volume and Total Bags.

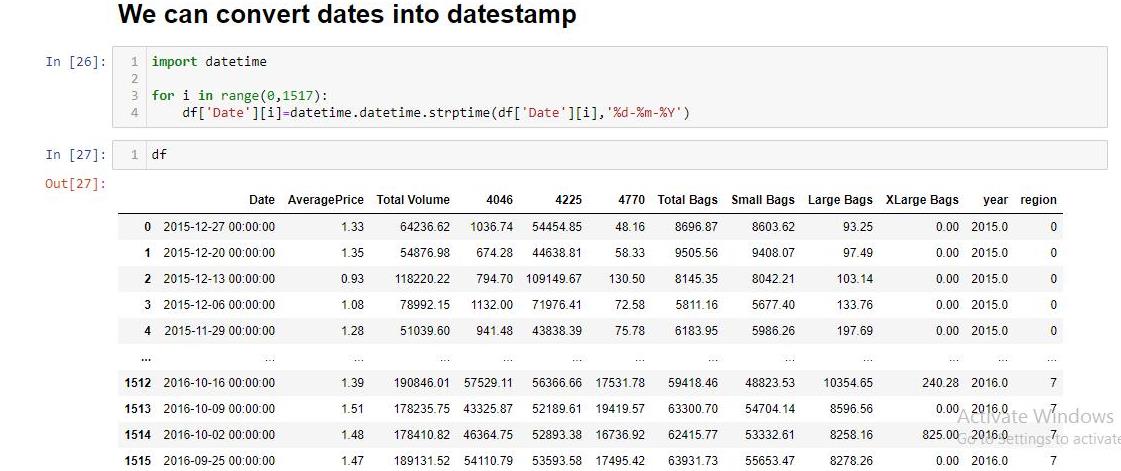


* The increase in volume increased the total no. of bags. It is a linear relationship.

Label Encoding is used to convert categorical data into numeric.

Now region is converted to numeric.

Converting the date data into datestamp as the it was in string type. Datetime is the library which is used for datestamp type datas.



Years are encoded by subtracting the oldest the year of the dataset from the current year. As there are two years in the dataset ie 2015 and 2016. So year is encoded as 0(2015) and 1(2016).



Now dropping the year column containing original data as 2015 and 2016.



Now Date is encoded as same as the year by subtracting the oldest date of the dataset from the current date.

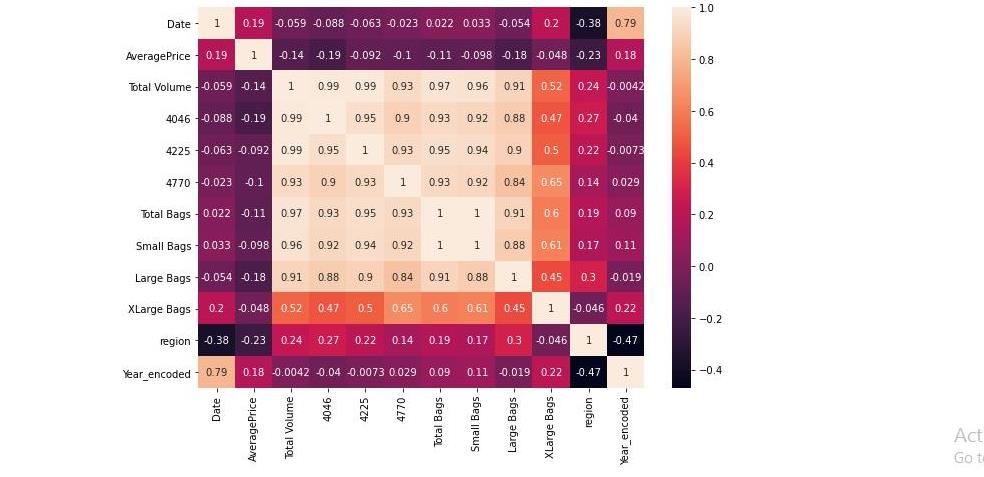
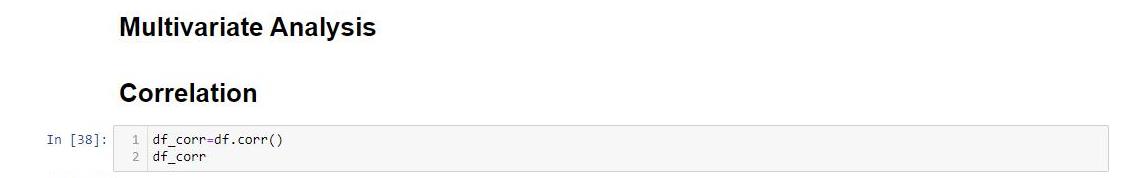


Now the Date\_encoded column is converted into numeric using (.days) extension with datestamp data. Then that is saved in Date column and the Date\_encoded column is dropped. So a lot of confusion in codes are entertained.





Now all the data is converted into numeric. And the relationship between all features can be shown in ***Multivariate*** Analysis using ***Correlation***.

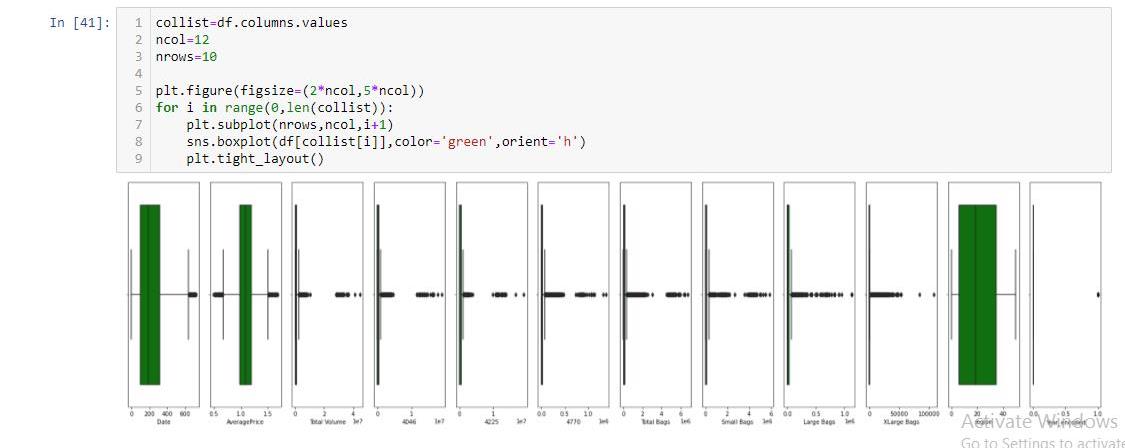


Averageprice is positively correlated with Year\_encoded and Date while negatively correlated with region,total bags,total volume.

Region is positively correlated with Bags,Volume while negatively correlated with Date,Year\_encoded and Averageprice.

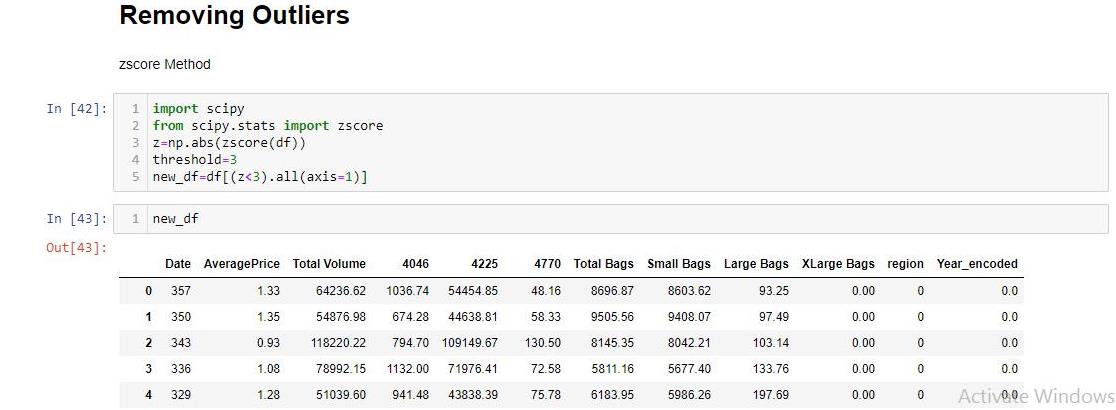
From this observation, we got some knowledge that it is possible to build prediction models for both AveragePrice and Region separately.

Now Checking for **outliers** present in the Dataset.

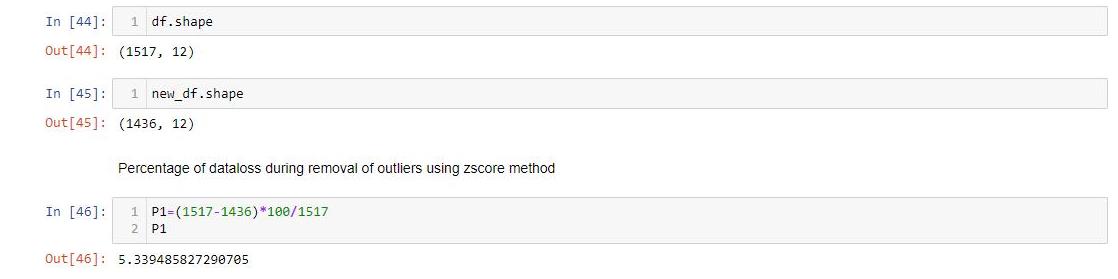


There so much outliers in all Volumes and Bags.

Now, Removing Outliers from the dataset using **zscore** method.

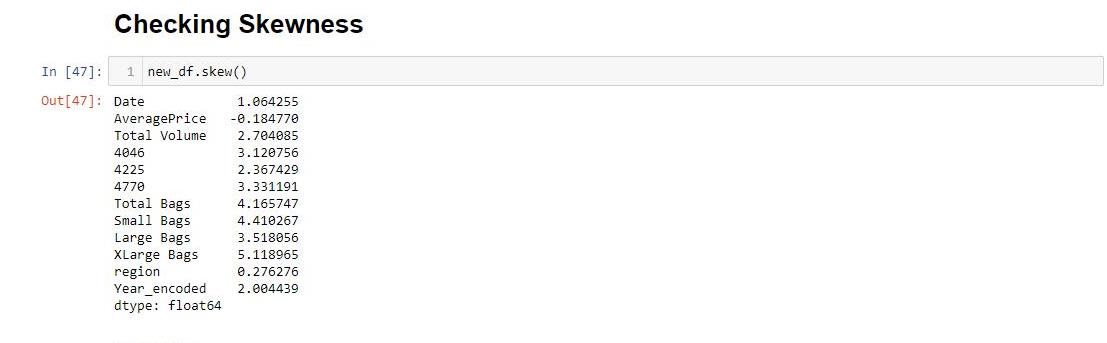


Comparing the new size of the new dataframe new\_df with old dataframe df after removing outliers.



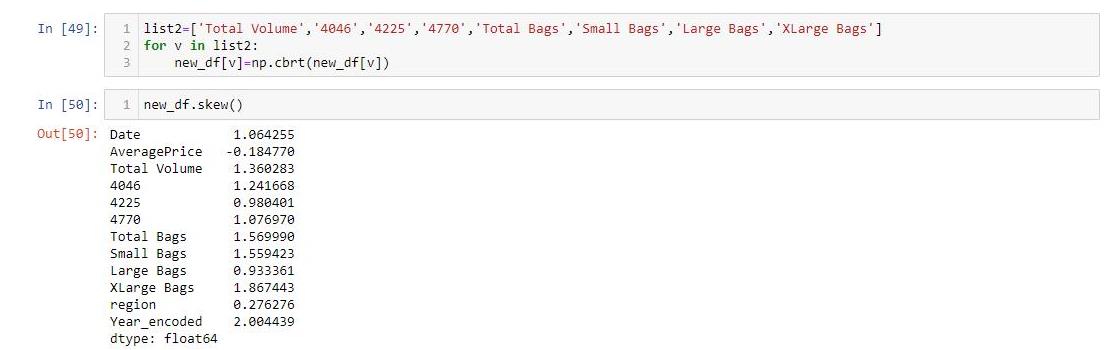
Percentage of data loss after removing outliers is 5.34% which is allowed to carry on with new dataframe ie new\_df.

Now Checking for **skewness** in the dataframe.

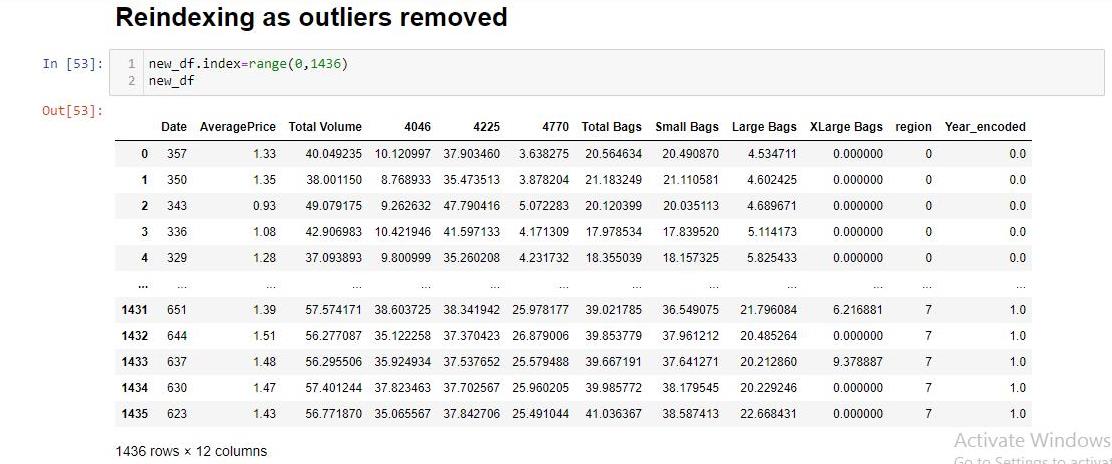


There are so much skewness in the dataset even after removing the outliers. The normal range for the distribution of datas is -0.5 to 0.5.

To remove the skewness here **cuberoot transformation** is used.



Now reindexing the new\_df.



All the EDA process is completed. The procedure was as follows:

* **Null values were removed.**
* **Data Visualization was done including Univariate, Bivariate and Multivariate Analysis.**
* **Dates are converted into Datestamps and encoded as days from the oldest date.**
* **Years are encodes as 0 and 1 for 2015 and 2016 respectively.**
* **All extra or unnecessary columns are dropped.**
* **All the outliers are removed.**
* **Skewness is reduced.**

Now the new\_df dataframe is ready for prediction models.

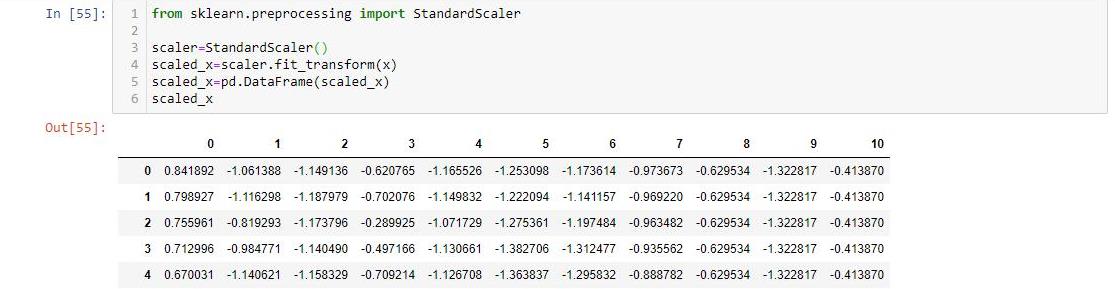
**Preprocessing for Building Models:**

The First Model(say Model 1) is designed to predict AveragePrice. Here, AveragePrice is the target variable.



Storing the input data in x variable. Here x is the independent variable.

Scaling the Input Data to fed it to the prediction model for better results.



Now the datas are scaled between -1 and 1.

Preparing the target variable and denoted as y. AveragePrice is stored in y.

This model is **regression** model as the AveragePrice is a continuous numeric data.



Now the preprocessing work is over. It is ready to fed in a model.

**Building a Regression Model for prediction of AveragePrice:**

Now, importing train\_test\_split, LinearRegression and r2\_score to find the best random state at which the model will perform the best.



The best random state is at 73.

All regression algorithms are imported such as **Linear Regression, KNeighbor Regression, SVR**.

Ensemble techniques such as **Random Forest Regressor, Gradient Boosting Regressor, AdaBoost Regressor** are also imported.

Regularization algorithms to avoid overfitting and underfitting such as **Lasso, Ridge, ElasticNet** are also imported.

cross\_val\_score is imported for **Cross Validation**.

**Metrics** for Regression are also imported. Such as:

**r2\_score**: It tells the accuracy of the model.

**mean\_squared\_error**: It tells the squared distances between the predicted values and actual values in the graphical representation of Input and Target Variables. The square root of mean\_squared\_error is taken as the metric value to check the performance of the model.

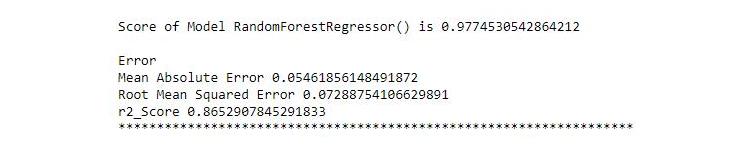
**mean\_absolute\_error**: It tells the average magnitudes of the distances between the predicted values and actual values in the graphical representation of Input and Target Variables.

Now, the Multiple Algorithms are used by a for loop to get the results at a glance.

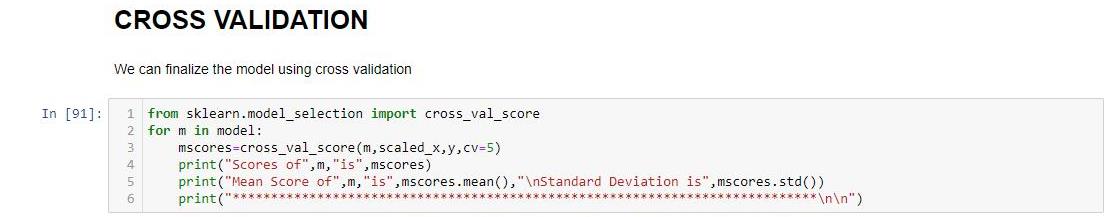


The algorithm that gives the best results for AveragePrice Prediction is **Random Forest Regressor**. The r2\_score which is also taken as accuracy of the model is 0.8653 means 86.52% is the Accuracy of the model.

The Mean Absolute Error and Root Mean Squared Error are close and it is 0.05-0.07. Means the error varies in AveragePrice is 0.05-0.07.



To improve the accuracy Cross Validation is used. But, the accuracy is not increased beyond 86.52% using cross validation in any model.

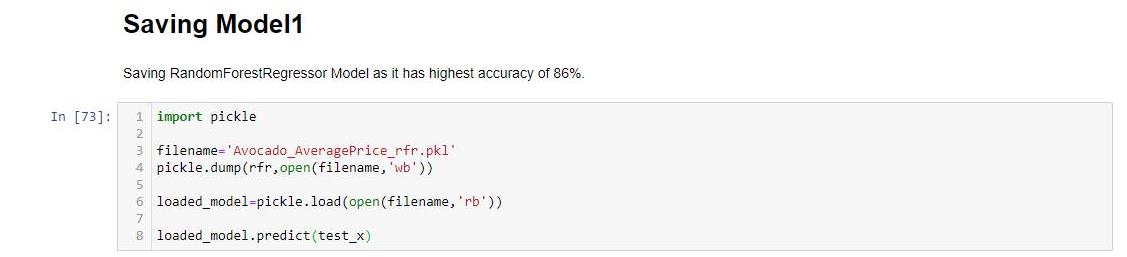


The **Hypertuning** of the best model Random Forest Regressor is done using **GridSearchCV**.



The accuracy of the model is slightly improved to 86.60% from 86.52%.

Now, saving this model of using pickle.



Now, Model1 is ready for prediction of AveragePrice.

Seeing the predicted result of the test\_x using the best saved model.



Saving the predicted result in a **CSV** file.



**Preprocessing of Model 2:**

Model 2 is built to predict Region. As Region is categorical variable, the model is a **classification** model.

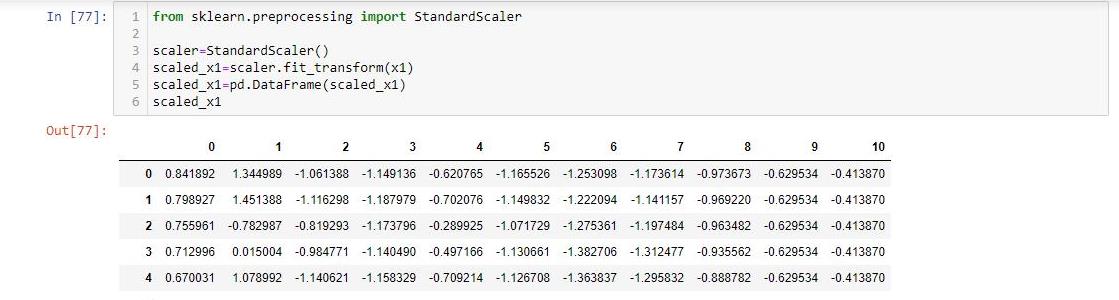
So Loading new\_df.



Preparing Input Variables by dropping region from the new\_df dataframe and it is denoted as x1.



Scaling the Input Variables in the scale of -1 to 1 using Standard Scaler.



Preparing Target Variable that contains only region column. It is denoted as y1.



**Building Classification Model for the Prediction of region:**

Finding the Best Random State for building the model to get best performance.



Best Random State is at 189.

All classifier algorithms are used.

The imported algorithms are **Logistic Regression, KNeighbor Classifier, Decision Tree Classifier, Support Vector Classifier(SVC)**.

Ensemble Techniques such as **Random Forest Classifier, AdaBoost Classifier, Gradient Boosting Classifier** are also imported.

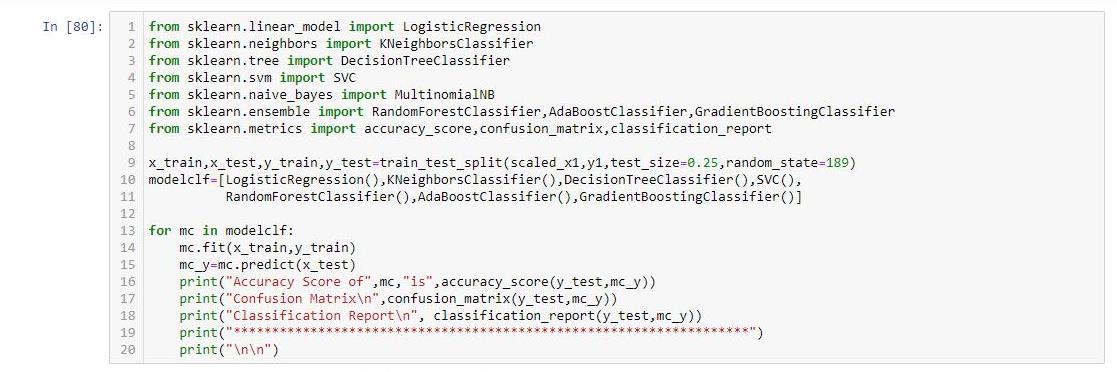
Some **Metrics** for classification are also imported such as **accuracy\_score, confusion\_matrix and classification\_report.**

**accuracy\_score** derives the accuracy of the model.

**confusion\_matrix** is the showcase to show the diagonal values are the true determined by both actual and prediction.

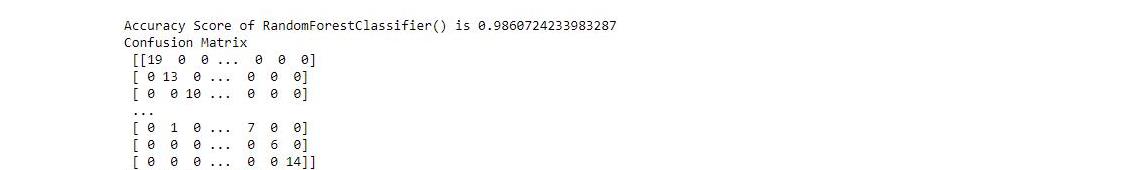
**classification\_report** tells the details of f1 score, precision, recall and support of each class of the target variable.

Now, the Multiple Algorithms are used by a for loop to get the results at a glance.



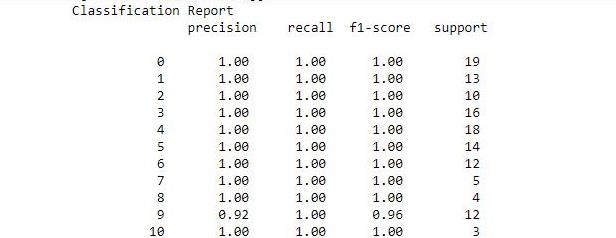
The best result, I got from Random Forest Classifier having accuracy\_score of 0.98607 means Accuracy of 98.61%.

The confusion Matrix tells maximum 0s in the matrix except the diagonal column which is actually true values determined by both actual n prediction model.

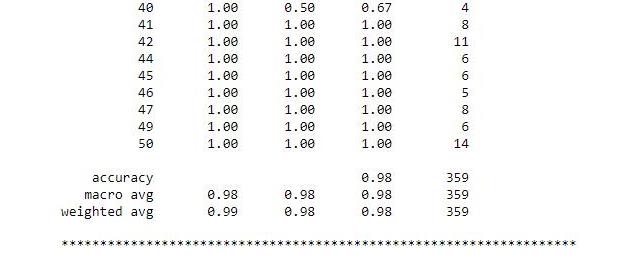


The Classification Report tells f1 score, recall n precision of all the classes of region. Those are close to 1, are giving best results for that class.

The accuracy of the model increases if all the class’s precision, recall and f1-score are close to 1.



-- --- --- --- --

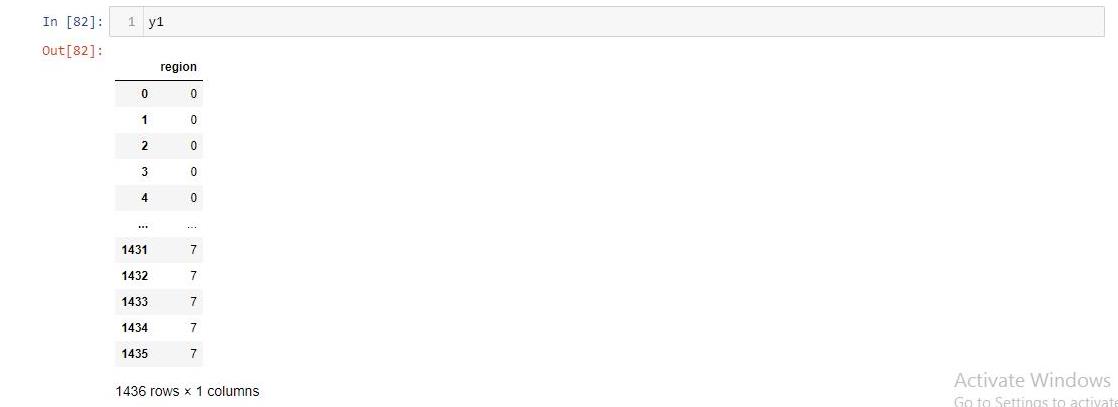


As MultinominalNB doesn’t support negative values, The input Variables are directly fed to the model without scaling.

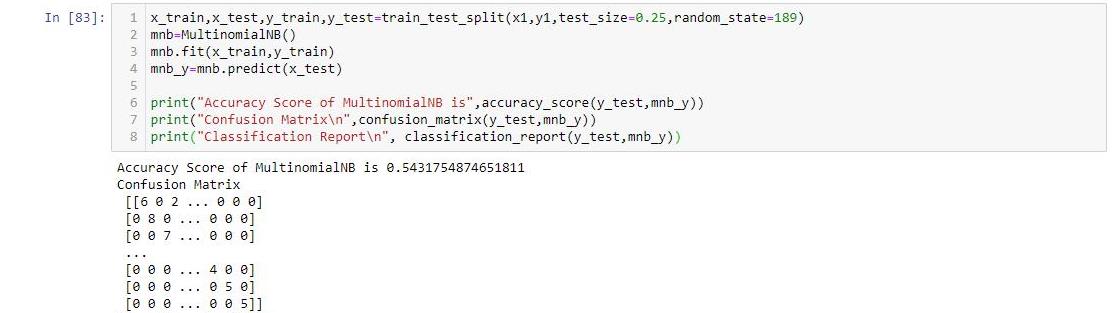
Calling the Input Variable that was stored in x1 before scaling.



Calling the Target Variable y1.



Both Input and Target Variables are fed to MultinomialNB algorithm.



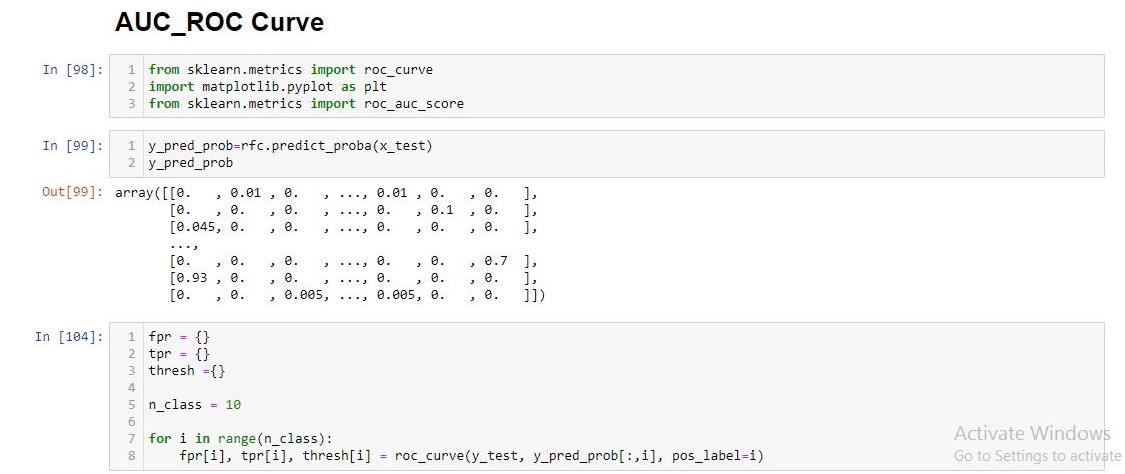
The Accuracy Score is very less (54.32%) as compared to other algorithms that has been fed after scaling.

Now, **Hypertuning** the best Model ie **Random Forest Classifier** which gave 98.61% Accuracy using **GridSearchCV.**



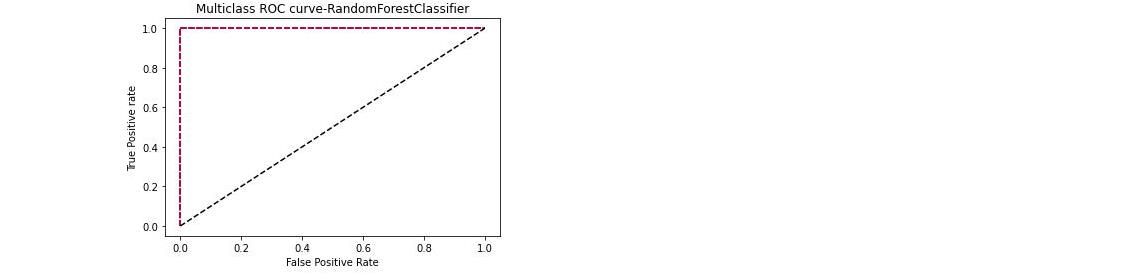
After the Hypertuning the Accuracy Score of Random Forest Classifier is same as before ie 98.61%. And this is the best Model.

Plotting AUC\_ROC Curve for first 10 classes of region using best model ie Random Forest Classifier.

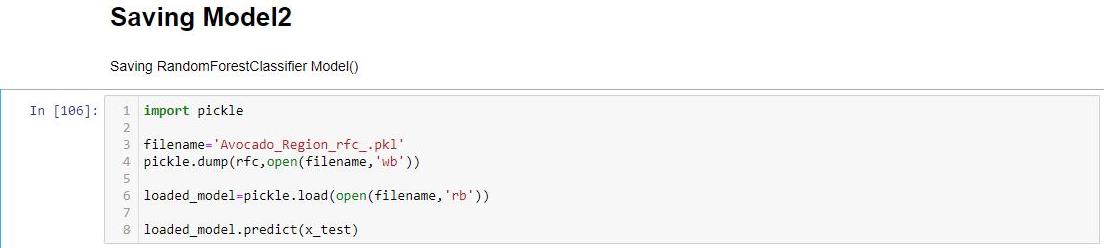




Almost all the 10 classes’ curves are similar and close to 1. And the Area under the curve is maximum.



Now, Saving the best Classication model for region prediction.



Predicted results obtained from x\_test.



Saving the results into a **CSV** file.



**Conclusion:**

* Now, there are two models saved. One for AveragePrice Prediction and another one for Region Prediction.
* Both models are made by using Random Forest Ensemble technique.
* AveragePrice Prediction Model is a Regression Model having accuracy of 86.61% whereas Region Prediction is a Classification Model having accuracy of 98.61%.

**Personal Experience:**

This Avocado Dataset contains a huge number of null values means there was no entry after 1517 rows. The EDA process was done easily. The only thing new was the datestamp. Its involvement was important to describe the data daywise. There were two Target Variables taken in different angles separately. Regression Model accuracy was a little bit low as 86% but Classification Model accuracy was 98%. Both Models were good and give close to the desired predicted results.

**Author**

**Aditya Prasad Pattanaik**